How and Why to Program Your Proofs

```
(as : List a) → (i : Nat) → i < length as → a
```

Gabriel Field

2025-Mar-07

Outline

Outline

The Colouring Problem

How many colours are needed to cover any map of countries such that no neighbouring same-coloured countries?

Figure: N00B5L4YER's `r/mapporncirclejerk` World Map [7]

The Four Colour Theorem

Theorem: Four colours suffice [2, 3].

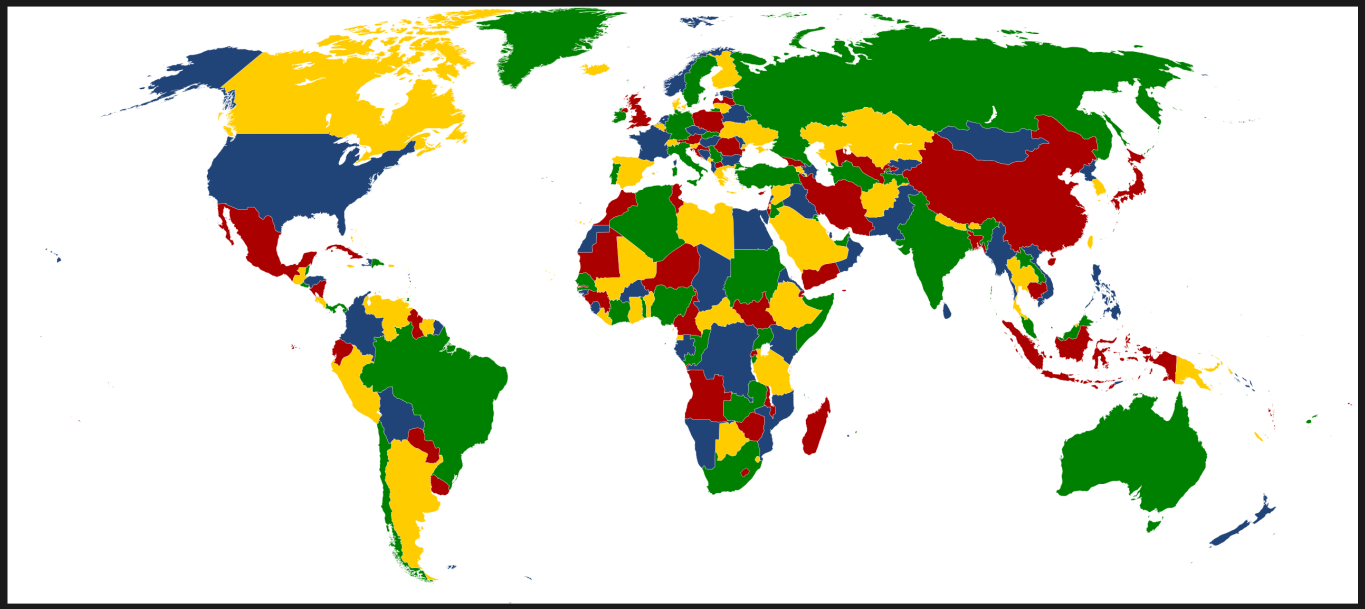

Figure: XaID's Four-Coloured World Map [8]

The Four Colour Theorem

Theorem: Four colours suffice [2, 3].

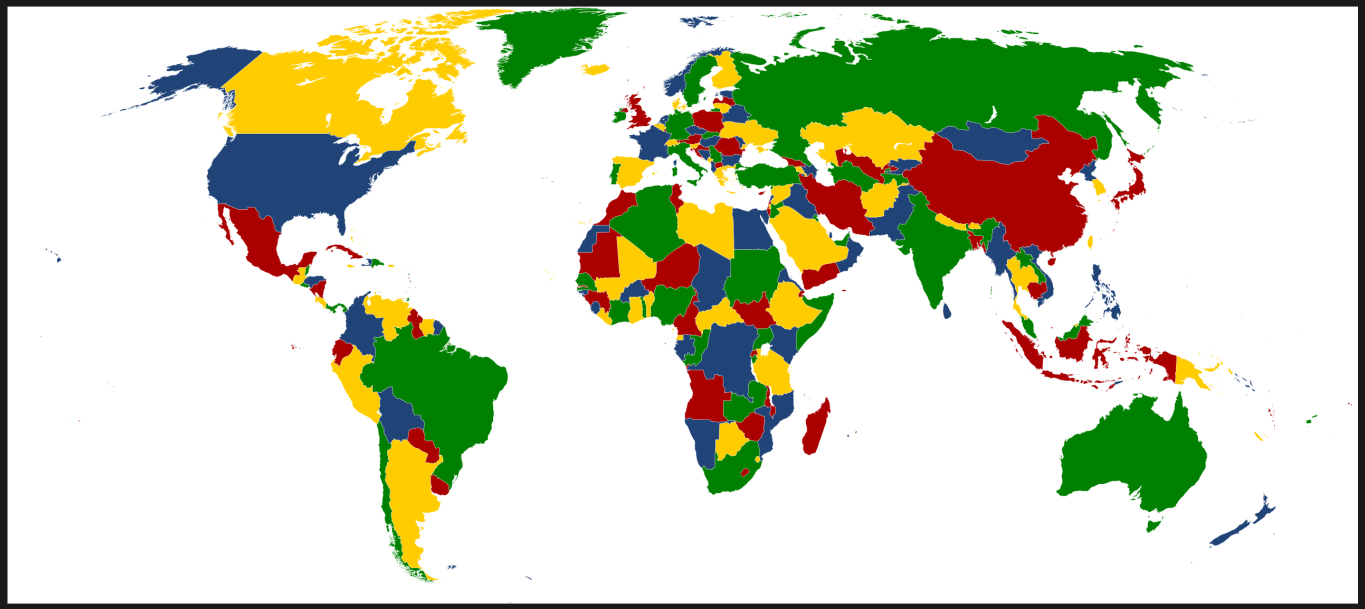

Figure: XaID's Four-Coloured World Map [8]

Proof: Get a computer to check **1834** distinct cases.

The Four Colour Theorem

Theorem: Four colours suffice [2, 3].

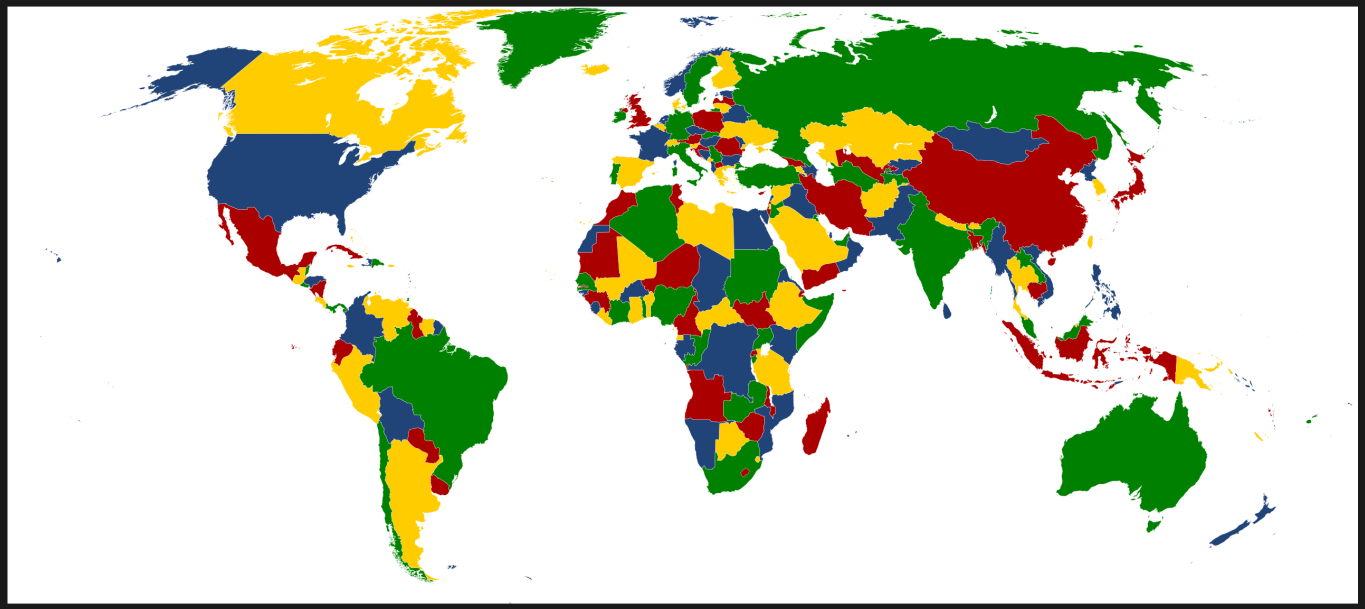

Figure: XaID's Four-Coloured World Map [8]

Proof: Get a computer to check **1834** distinct cases.

The Four Colour Theorem Problem

Proof: Get a computer to check **1834** distinct cases.

The Four Colour Theorem Problem

Proof: Get a computer to check **1834** distinct cases.

Problem: How do we check this proof?!

The Four Colour Theorem Problem

Proof: Get a computer to check **1834** distinct cases.

Problem: How do we check this proof?!

- How do we *practically* check 1834 different cases?

The Four Colour Theorem Problem

Proof: Get a computer to check **1834** distinct cases.

Problem: How do we check this proof?!

- How do we *practically* check 1834 different cases?
- How do we check that *software is correct*?

The Four Colour Theorem Problem

Proof: Get a computer to check **1834** distinct cases.

Problem: How do we check this proof?!

- How do we *practically* check 1834 different cases?
- How do we check that *software is correct*?

Solution: Write a smaller program to check that the software is correct, and manually prove your smaller program correct.

The Four Colour Theorem Problem

Proof: Get a computer to check **1834** distinct cases.

Problem: How do we check this proof?!

- How do we *practically* check 1834 different cases?
- How do we check that *software is correct*?

Solution: Write a smaller program to check that the software is correct, and manually prove your smaller program correct.

This smaller program is a *proof assistant*.

Proof Assistants

Proof assistants help with:

Proof Assistants

Proof assistants help with:

- Catching human mistakes

Proof Assistants

Proof assistants help with:

- Catching human mistakes
- Verifying critical software ("formal verification")

Proof Assistants

Proof assistants help with:

- Catching human mistakes
- Verifying critical software ("formal verification")

Proof *automation* helps with:

Proof Assistants

Proof assistants help with:

- Catching human mistakes
- Verifying critical software ("formal verification")

Proof *automation* helps with:

- Menial proofs (e.g. 1834 cases)

Proof Assistants

Proof assistants help with:

- Catching human mistakes
- Verifying critical software ("formal verification")

Proof *automation* helps with:

- Menial proofs (e.g. 1834 cases)
- Accessibility of formal verification ($\Rightarrow$ *safer* code)

Proof Assistants

Proof assistants help with:

- Catching human mistakes
- Verifying critical software ("formal verification")

Proof *automation* helps with:

- Menial proofs (e.g. 1834 cases)
- Accessibility of formal verification ($\Rightarrow$ *safer* code)

Figure: The NASA Langley Formal Methods Research Program [1].

Proof Assistants

Proof assistants help with:

- Catching human mistakes
- Verifying critical software ("formal verification")

Proof *automation* helps with:

- Menial proofs (e.g. 1834 cases)
- Accessibility of formal verification ($\Rightarrow$ *safer* code)

Figure: The Dafny Programming and Verification Language [4].

Proof Assistants

Proof assistants help with:

- Catching human mistakes
- Verifying critical software ("formal verification")

Proof *automation* helps with:

- Menial proofs (e.g. 1834 cases)
- Accessibility of formal verification ($\Rightarrow$ *safer* code)

```
1  Theorem four_color m : simple_map m -> colorable_with 4 m.
2  Proof.
3    revert m;
4    exact (finitize.compactness_extension four_color_finite).
5    Qed.
```

Figure: Gonthier's verified proof of the Four Colour Theorem [5, 6].

How Do They Work?

So, how do proof assistants work?

Outline

How Did They Work?

How Did They Work?

```
proof : statement.
```

How Did They Work?

`proof : statement.`

Induction is recursion.

How Did They Work?

`proof : statement`.

Induction is recursion.

Automation makes this paradigm practical.

Thx 4 watch

Thanks! (let's have pizza)

References.

[1] National Aeronautics and Space Administration. *NASA Langley Formal Methods Research Program*. 2024. URL: https://shemesh.larc.nasa.gov/fm/.

[2] K. Appel and W. Haken. "Every planar map is four colorable. Part I: Discharging". In: *Illinois Journal of Mathematics* 21.3 (1977), pp. 429–490. DOI: 10.1215/ijm/1256049011. URL: https://doi.org/10.1215/ijm/1256049011.

[3] K. Appel, W. Haken, and J. Koch. "Every planar map is four colorable. Part II: Reducibility". In: *Illinois Journal of Mathematics* 21.3 (1977), pp. 491–567. DOI: 10.1215/ijm/1256049012. URL: https://doi.org/10.1215/ijm/1256049012.

[4] Dafny. *The Dafny Programming and Verification Language*. 2025. URL: https://dafny.org/.

[5] G. Gonthier. "Formal Proof—The Four-Color Theorem". In: *Notices of the AMS* 55 (2008), pp. 1382–1393. URL: http://www.ams.org/notices/200811/tx081101382p.pdf.

[6] G. Gonthier and Yves Bertot. *The Four Colour Theorem*. 2024. URL: https://github.com/coq-community/fourcolor.

[7] N00B5L4YER. *map of the world but most voted comments change nothing cuz the world is perfect as it is, perfectly imperfect, whether u like it or not. also dark mode*. 2023. URL: https://www.reddit.com/r/mapporncirclejerk/comments/10jl0wq/map_of_the_world_but_most_voted_comments_change/.

[8] XalD. *File:Four color world map.svg*. 2023. URL: https://upload.wikimedia.org/wikipedia/commons/thumb/0/0a/Four_color_world_map.svg/2560px-Four_color_world_map.svg.png.